



## 成都远向电子科技有限公司产品说明书

产品型号：YX-G4AI-485

全部资料下载地址：<http://ask.zstel.com:8090>

技术支持服务电话：028-64267900

技术支持专员企业 QQ：3183329475

官网网站：<https://www.zstel.com/>

硬件/软件技术定制热线：19150158475 张工

# 目录

目录.....	2
一、产品概述.....	3
1.1 概述.....	3
1.2 性能特点.....	3
1.3 技术参数.....	3
二、外观尺寸.....	4
2.1 产品外观.....	4
2.2 产品尺寸图.....	5
三、产品接线图、跳线、指示灯说明.....	6
3.1 端子描述.....	6
3.2 跳线.....	7
3.3 LED 指示灯.....	7
3.4 接线图.....	7
四、ModbusRTU 通讯协议、组态软件软件说明.....	9
4.1 通讯协议.....	9
4.2 寄存器地址.....	9
4.2 Modbus RTU 功能码.....	9
4.3 Modbus 通讯实例.....	10
五、软件操作.....	12
5.1 配置软件.....	12
5.2 配置基本参数.....	13
5.3 DO 继电器输出相关参数.....	13
5.4 AI 模拟量采集相关参数.....	14
5.5 模拟量通道校准相关参数.....	14
5.6 其他功能.....	16
六、协议详解.....	22
6.1 功能码描述.....	22
6.2 错误码描述.....	28
6.3 CRC 校验算法.....	29
七、更改记录.....	31

# 一、产品概述

## 1.1 概述

YX-G4AI-485 是一款工业级标准模拟量采集产品，共有 4 个测量通道，每个通道均可以分别设置多种量程，同时带有 4 路常开继电器输出，可实现 AI-D0 联动；RS-485 通讯接口使用标准 Modbus RTU 协议，符合工业标准。

## 1.2 性能特点

- 防死机硬件看门狗
- 5~35V 带防反接、过压过流保护电源
- 带模拟量通道隔离 4 路模拟量电流输入 0~20mA，电压输入 0~5V
- 4 路继电器常开输出
- 12 位分辨率，0.1%精度 ADC
- 高性能低功耗 32 位 ARM 嵌入式 CPU
- 支持 ModbusRTU 从站协议
- 1 路指示灯
- 带防雷、静电保护、电源隔离 RS485 通讯接口
- 工业温度范围，应对严苛现场环境
- 自定义线性模拟量数据转换

## 1.3 技术参数

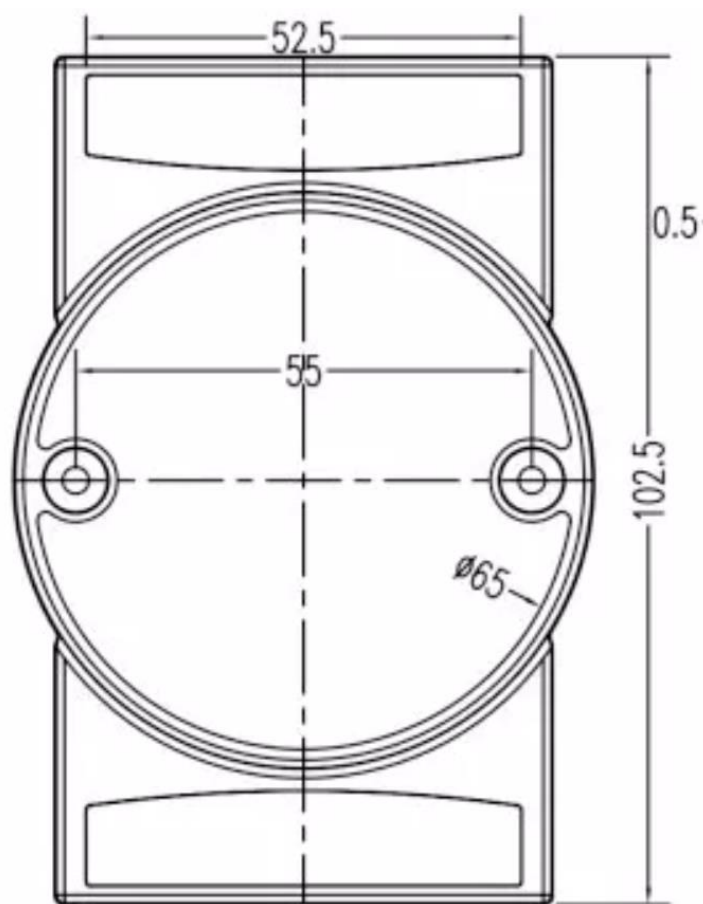
模拟量接口	AI	4 路双端
	AI 分辨率	12bit
	AI 量程	0~5V、0~20mA
	精度	0.1%
	采集速度	单通道固定 1KHz
	AI 输入阻抗	0~20mA $\leq 120\Omega$ 0~5V $\geq 10K\Omega$
开关量输出	继电器类型	4 路继电器 常开
	触电电阻	100m $\Omega$
	机械寿命	1x10 <sup>6</sup> 次
	最大切换电压	0~250VAC，0~30VDC
	最大切换电流	0~5A
通讯接口	通讯接口	RS485
	波特率	1200~115200bps
	数据格式	8N1, 8E1, 8O1, 8N2, 8E2, 8O2
	通讯协议	ModbusRTU
	过压过流保护	45V
电源参数	电源规格	DC 5~36V
	功耗	12V-0.5W
工作环境	工作温度、湿度	-40℃~85℃，0%RH~95%RH
其他	尺寸	82*50*32

## 二、外观尺寸

### 2.1 产品外观

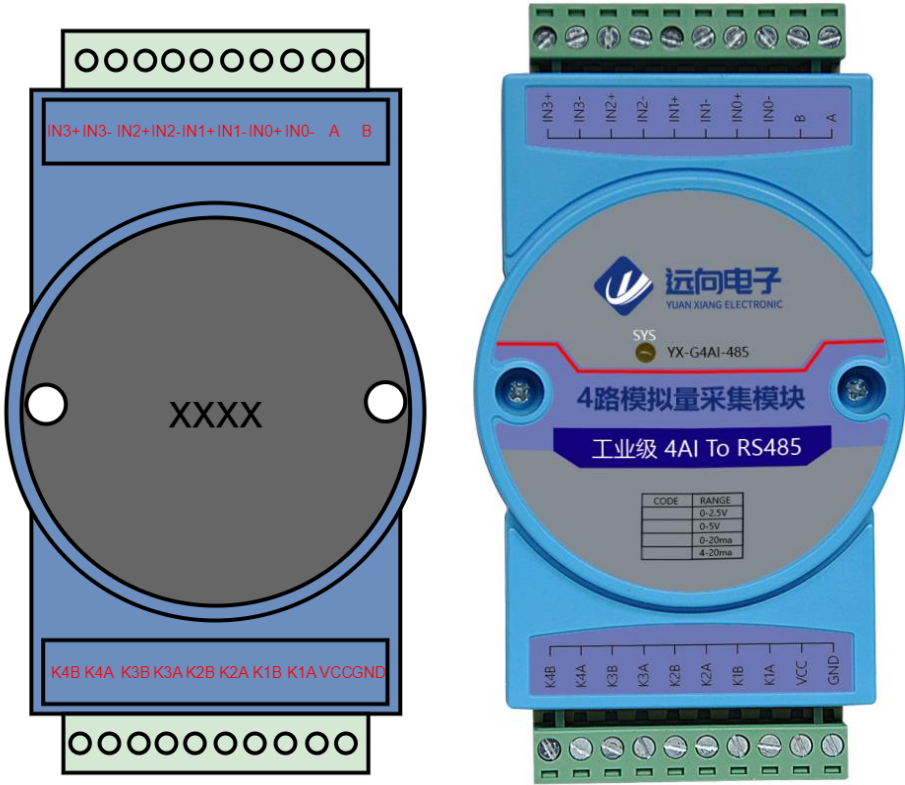


## 2.2 产品尺寸图



### 三、产品接线图、跳线、指示灯说明

#### 3.1 端子描述



顶部 10 槽接线位：

- A: RS485 通讯 A
- B: RS485 通讯 B
- IN0+: 模拟量输入通道 0 正极
- IN0-: 模拟量输入通道 0 负极
- IN1+: 模拟量输入通道 1 正极

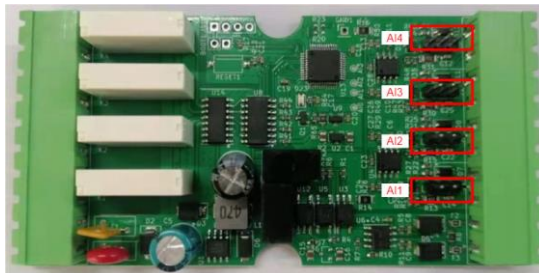
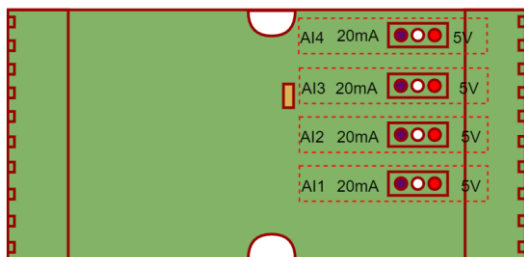
- IN1-: 模拟量输入通道 1 负极
- IN2+: 模拟量输入通道 2 正极
- IN2-: 模拟量输入通道 2 负极
- IN3+: 模拟量输入通道 3 正极
- IN3-: 模拟量输入通道 3 负极

底部 10 槽接线位：

- VCC: 电源正极
- GND: 电源负极
- K1A: 开关量输出 D01 进线端
- K1B: 开关量输出 D01 出线端
- K2A: 开关量输出 D02 进线端

- K2A: 开关量输出 D02 出线端
- K3A: 开关量输出 D03 进线端
- K3A: 开关量输出 D03 出线端
- K4A: 开关量输出 D04 进线端
- K4A: 开关量输出 D04 出线端

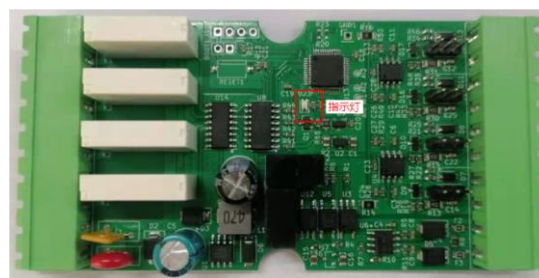
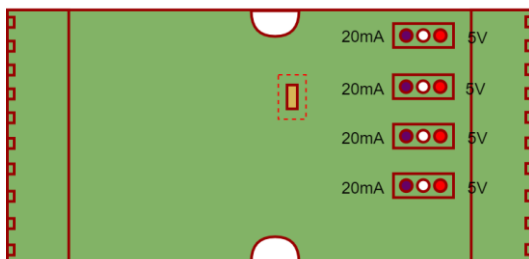
### 3.2 跳线



电路板上 8 个跳线座，通过跳线选择不同点量程，结合配置软件实现多量程功能：

- 通道  $AI_x$ ：
  - 0~5V 量程：选择红色外侧 2 个跳线座
  - 0~20mA 量程：选择紫色内侧 2 个跳线座

### 3.3 LED 指示灯



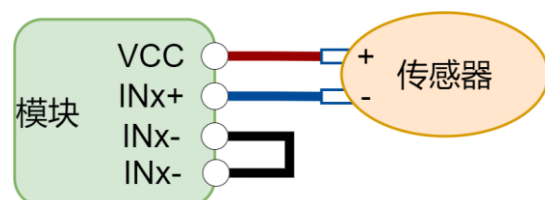
1 个 LED 指示灯：

- **SYS 指示灯**：每 1 秒闪烁 1 次，表示正常工作

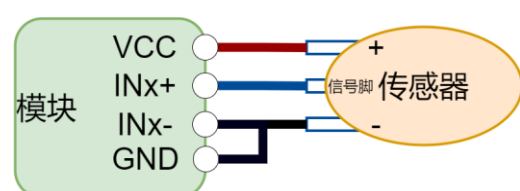
### 3.4 接线图

(1) 模拟量(AI)输入接线：

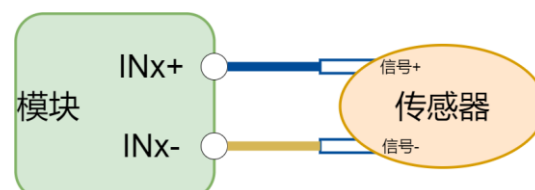
无源传感器（2 线制）：



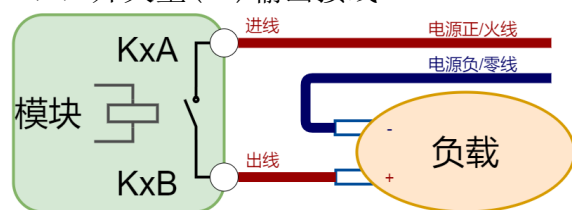
有源传感器（3 线制）：



有源传感器（2/4 线制）：



(2) 开关量(D0)输出接线



四、ModbusRTU 通讯协议、组态软件软件说明

4.1 通讯协议

本产品兼容标准 Modbus RTU 从站协议，能够支持标准 Modbus RTU 组态软件，详细介绍参考本文第六章内容  
快速上手可参照“[4.3 Modbus 通讯实例](#)”

4.2 寄存器地址

寄存器地址	名称	字节数	说明	备注
模拟量输入				
0x0000 (0)	AI1_H	2	模拟量通道 1 高	每个模拟量通道占 2 个 Modbus 寄存器，4 个字节，格式为浮点数，浮点数格式符合 IEEE 754 标准 可参照 5.3 读取 AI 0~20mA：0~20.0 0~5V：0~5.0
0x0001 (1)	AI1_L	2	模拟量通道 1 低	
0x0002 (2)	AI2_H	2	模拟量通道 2 高	
0x0003 (3)	AI2_L	2	模拟量通道 2 低	
0x0004 (4)	AI3_H	2	模拟量通道 3 高	
0x0005 (5)	AI3_L	2	模拟量通道 3 低	
0x0006 (6)	AI4_H	2	模拟量通道 4 高	
0x0007 (7)	AI4_L	2	模拟量通道 4 低	
0x0100 (256)	AI1_D	2	模拟量通道 1	AI 整数寄存器 0~20mA：0~2000 0~5V：0~500
0x0101 (257)	AI2_D	2	模拟量通道 2	
0x0100 (258)	AI3_D	2	模拟量通道 3	
0x0101 (259)	AI4_D	2	模拟量通道 4	
开关量输出				
0x0014 (20)	D01	2	开关量通道输出 1	开关量输出 继电器控制
0x0015 (21)	D02	2	开关量通道输出 2	
0x0016 (22)	D03	2	开关量通道输出 3	
0x0017 (23)	D04	2	开关量通道输出 4	

4.2 Modbus RTU 功能码

功能码	操作	说明
01	读取单位 DO 状态	Bit 位表示 DO 输出状态

03	读取 AI, D0 寄存器值	读取 AI, D0 寄存器值
04	读取 AI, D0 寄存器值	读取 AI, D0 寄存器值
05	写单个 D0	0xFF00: 闭合;0x0000: 断开
06	写单个 D0	0x0001: 闭合;0x0000: 断开
0F	写多个 D0	参照本文第六章内容
10	写多个 D0	参照本文第六章内容

详细讲解参照本文第六章内容

### 4.3 Modbus 通讯实例

#### (1) 读取 D01:

##### a. 用 01 功能码读取 D01

发送: 01 01 00 14 00 01 BD CE

接受: 01 01 01 00 51 88

##### b. 用 03 功能码读取 D01

发送: 01 03 00 14 00 01 C4 0E

接受: 01 03 02 00 00 B8 44

##### c. 用 04 功能码读取 D01

发送: 01 04 00 14 00 01 71 CE

接受: 01 04 02 00 00 B9 30

#### (2) 操作 D01:

##### a. 用 05 功能码操作单个 D01

发送: 01 05 00 14 FF 00 CC 3E

接受: 01 05 00 14 FF 00 CC 3E

##### b. 用 06 功能码操作单个 D01

发送: 01 06 00 14 00 01 08 0E

接受: 01 06 00 14 00 01 08 0E

##### c. 用 0F 功能码操作多个 D01、D02

发送: 01 0F 00 14 00 02 01 03 AE 95

接受: 01 0F 00 14 00 02 94 0E

d. 用 10 功能码操作多个 D01、D02

发送: 01 10 00 14 00 02 04 00 01 00 01 63 50  
接受: 01 10 00 14 00 02 01 CC

### (3) 读取 AI:

给定输入 4.96 (40 9E E7 CF)

a. 用 03 功能码读取浮点数 AI1:

发送: 01 03 00 00 00 02 C4 0B  
接受: 01 03 04 40 9E E7 CF 85 B9  
4.96 IEE 浮点数十六进制为 (40 9E E7 CF)

b. 用 04 功能码读取浮点数 AI1:

发送: 01 04 00 00 00 02 71 CB  
接受: 01 04 04 40 9E CE 1F 9A 02

C. 用 03 功能码读取整数 AI1:

发送: 01 03 01 00 00 01 85 F6  
接收: 01 03 02 01 F0 B9 90  
整数读出数值为 496 (0x01F0)

d. 用 04 功能码读取整数 AI1:

发送: 01 04 01 00 00 01 30 36  
接收: 01 04 02 01 F0 B8 E4  
整数读出数值为 496 (0x01F0)

# 五、软件操作

设备参数配置教程，结合《用户测试文档》即可对设备进行简单测试

## 5.1 配置软件

参数配置软件介绍：



5.1.1 配置软件包含有：

- **功能区**：包含有配置软件所支持功能, 以及功能那个切换选项
- **参数配置主区域**：参数配置主要区域，参数项的读取、写入临时列表
- **串口/命令集区**：涉及模块的参数读、写、重启等操作
- **串口日志区**：命令集的操作日志

5.1.2 参数配置准备：

- (1) 用 USB-485 工具连接设备到电脑
- (2) 在串口配置框内配置串口波特率、停止位、校验位、数据位；（默认波特率 9600，数据位 8，停止位 1，校验位 None）
- (3) 选择串口配置框子项 “**命令集**”



- (4) 点击“**读取参数**”命令按钮，读取设备参数（不同设备拥有不同指令集）
- (5) 双击对应参数项的“**参数值**”，然后对参数进行修改
- (6) 修改完参数后需要点击命令集里的“**设置参数**”，写入到模块中
- (7) 写入完成在日志区域会提示成功。



- (8) 通过点击“**重启设备**”按钮，重启模块设备使配置参数生效

## 5.2 配置基本参数

该系列参数涉及到对 485 通讯 Modbus 协议相关配置。

参数名称	参数值	参数说明
<基本参数>		
Modbus地址	<b>双击修改参数</b>	设备的Modbus地址, 1~255
通信模块波特率		与通信模块的波特率一致, 一般设置为9600
通信模块奇偶校验		与通信模块的奇偶校验一致, 一般设置为8N1

- **Modbus 地址**: Modbus 地址参数, 可设置 1~255
- **通讯模块波特率**: 设备 485 通讯波特率 (波特率支持主流的波特率选项)
- **通讯模块就校验**: 设备 485 通讯奇偶校验位, 可配置 8N1, 8E1, 8O1...

## 5.3 D0 继电器输出相关参数

部分产品包含有多个 D0 输出或者不包含有 D0 输出功能, 具体请根据实际配置软件显示栏目进行配置。D0 功能测试可以参考《**用户测试文档**》。

<DO继电器输出相关参数>		
DO1初始值	双击修改参数	定义第1路继电器的初始状态
DO2初始值		定义第2路继电器的初始状态

- **DOx 初始值：**通过选择参数可配置开机上电后 DO 输出状态默认为“等待上位操作”状态。

## 5.4 AI 模拟量采集相关参数

本系列参数涉及到对 AI 量程、AI 告警功能的配置。每个通道都包含如下的配置项。

<AI模拟量采集相关参数>		
AI1采集量程	双击修改参数	模拟量采集的量程
AI1告警下限值		浮点数,采集到的模拟量低于此值时, RTU告警
AI1告警上限值		浮点数,采集到的模拟量高于此值时, RTU告警
AI1告警操作DO端口		告警切换DO端口输出状态

- **AIx 采集量程：**选择对应选项修改 AI 采集量程
- **AIx 告警下限值：**采集数据低于此值时会触发 AIx 读数低于下限值告警事件，告警事件可用于 DO，告警消息联动功能
- **AIx 告警上限值：**采集数据高于此值时会触发 AIx 读数高于于下限值告警事件，告警事件可用于 DO，告警消息联动功能
- **AIx 告警操作 DO 端口：**当模拟量告警事件发生后，触发对 DO 的状态切换（由“低/断开”切换为“高/闭合”状态）

## 5.5 模拟量通道校准相关参数

本系列参数用于模拟量读数转换，RTU 采集模块默认读取出来的读数值为采集的电流/电压值，接传感器时会有事会用到将电流/电压值转换为实际传感器读数。则可以使用本系列参数实现读数转换功能。

<AI模拟通道校准相关参数>		
AI1增益值 gain		AI1采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$
AI1比例值 ratio	双击修改参数	AI1采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$
AI1偏移值 offset		AI1采集的模拟量计算公式 $AI = (adc + gain) * ratio + offset$

- **AIx 增益值 gain：**AIx 数值转换公式中的增益值(详细使用参考后续说明)
- **AIx 比例值 ratio：**AIx 数值转换公式中的比例值(详细使用参考后续说明)
- **AIx 偏移值 offset：**AIx 数值转换公式中的偏移值(详细使用参考后续说明)

### 5.5.1 转换原理讲解

模拟量读数转换公式如下（例：将电压转换为温度值）

$$AI = (adc + gain) \times ratio + offset$$

AIx 即转换后数值：

adc 为转换前读数值  
gain 为设定增益系数（初始为 0.0）  
ratio 为设定比例系数（初始为 1.0）  
offset 为设定位置系数（初始为 0.0）

$$ratio = \frac{\text{传感器}_{max} - \text{传感器}_{min}}{\text{模拟量}_{max} - \text{模拟量}_{min}}$$

$$offset = \frac{\text{模拟量}_{max} * \text{传感器}_{min} - \text{模拟量}_{min} * \text{传感器}_{max}}{\text{模拟量}_{max} - \text{模拟量}_{min}}$$

### 5.5.2 校准实例

本案例举例说明如何将一个 4~20mA 对应 -40℃~120℃ 的温度传感器输出的电流值转换为实际温度读数。

- ◆ 已知温度传感器 输出模拟量范围是 4~20mA，则将 RTU 采集模块量程设置为 0/4~20mA 量程范围。
- ◆ 计算需要的增益（gain）、比例（ratio）、偏移（offset）三个参数值。根据传感器参数可知

$$\text{模拟量}_{min} = 4$$

$$\text{模拟量}_{max} = 20$$

$$\text{传感器}_{min} = -40$$

$$\text{传感器}_{max} = 120$$

代入 ratio、offset 计算公式中可得

$$ratio = [120 - (-40)] / (20 - 4) = 10$$

$$offset = \{[20 * (-40)] - [4 * (120)]\} / (20 - 40) = -80$$

- ◆ 根据将计算后的参数代入转换公式即可得  
传感器输出模拟量 4mA ---> 传感器温度读数 = 4\*10 - 80 = -40℃  
传感器输出模拟量 20mA --> 传感器温度读数 = 20\*10 - 80 = 120℃

# 5.6 其他功能

## 5.6.1 校准模拟量精度

当系统精度不够时，产品提供一个用户精度校准功能

(1) 设备串口连接上电脑，点击“命令集”中“校准模拟量精度”按钮进行模拟量校准



(2) 取消勾选显示区“HEX”复选框，等到接受区提示如下内容时执行下一步



(3) 给定模拟量 CHx 通道恒定基准信号（注意通道量程）

(4) 发送通道基准信号格式 “CHx: xx”（例：“CH1: 10”）注意取消勾选发送

区“HEX”复选按钮  
(5) 等待设备提示校准消息

数据收发区

已接收:4819字节 收清零 已发送:189字节 发清零

【14:20:24.693】 下行:CH1: 10  
【成功】  
【14:20:31.317】 上行:■U  
【14:20:36.473】 上行:  
校准系数: 1.0000, 1.0002  
串口等待接受校准模拟量数值  
串口发送格式如下:  
CH1: xx  
CH2: xx  
【14:20:38.078】 下行:CH1: 10  
【14:20:38.238】 上行:模拟通道1校准完成  
【14:20:38.400】 上行:模拟通道1系数: 0.9986  
【14:20:38.597】 上行:退出模拟量校准模式  
【重启设备】  
【14:20:45.546】 下行:AA550004E00600EA  
【成功】  
【14:20:45.719】 上行:■U  
【成功】  
【14:20:46.325】 上行:HIGH

☐ HEX ☒ 显示时间 ☒ 自动换行 清空接受区

CH1: 10

☐ HEX ☒ 显示发送 ☐ 自动发送 清空发送区 发送

(6) 重启设备

5.6.2 AI-DO 联动

注意： 部分产品不包含 DO 功能即无 AI-DO 联动功能，实际请根据所购买的模块进行测试。

(1) 必要参数

<DO继电器输出相关参数>		
DO1初始值		定义第1路继电器的初始状态
<AI模拟量采集相关参数>		
AI1采集量程		模拟量采集的量程
AI1告警下限值		浮点数,采集到的模拟量低于此值时, RTU告警
AI1告警上限值		浮点数,采集到的模拟量高于此值时, RTU告警
AI1告警操作DO端口		告警切换DO端口输出状态

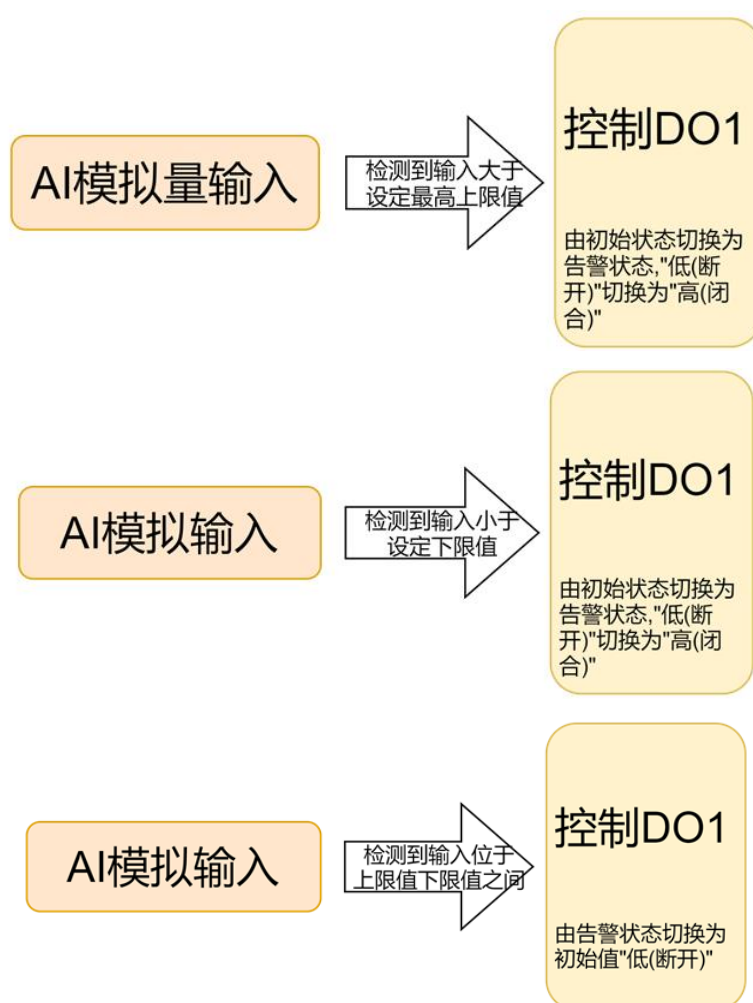
- DOx 初始值：配置正常空闲状态 DOx 状态
- AIx 采集量程： 模拟量输入的量程范围
- AIx 告警下限值： 设置 AIx 最低触发值，若不用可设置为模拟量最小值以下（4~20mA，设置的值小于 4mA 即用不能达到）
- AIx 告警上限值： 设置 AIx 最高触发值，若不使用上限值可设置大于最大量程值即不会触发。

(2) 实际案例  
配置如下参数：

DO1 初始值：低（断开）  
 AI1 采集量程：电流 0~20mA  
 AI1 告警下限值：1.0  
 AI1 告警上限值：15.0  
 AI1 告警操作 DO 端口：DO1

<DO继电器输出相关参数>		
DO1初始值	低（断开）	定义第1路继电器的初始状态
<AI模拟量采集相关参数>		
AI1采集量程	电流 0 ~ 20mA	模拟量采集的量程
AI1告警下限值	1.0	浮点数,采集到的模拟量低于此值时, RTU告警
AI1告警上限值	15.0	浮点数,采集到的模拟量高于此值时, RTU告警
AI1告警操作DO端口	DO1	告警切换DO端口输出状态

告警触发流程如下



### 5.6.3 消息告警内容参数配置(无线版)

#### (1) 必要参数

注：需要配合本公司远程模块, 485 有线版不开放该功能

- 安装地址： 主要用于报警通知使用（设备需包含有短信模块）
- 设备身份 ID： 主要用于报警通知使用（设备需包含有短信模块）
- AIx 告警周期： 用于重复发送告警消息
- AIx 低于下限值告警短信内容： AI 告警内容
- AIx 恢复短信内容： AI 告警内容
- AIx 超出上限告警短信内容： AI 告警内容

<短信告警内容参数>		当告警通道设置为短信时，此项参数才有效
安装地址		最长16个汉字,32个字符，短信报警用
设备身份ID		8位编码，短信报警时有效
AI1告警周期		循环告警的时间间隔，单位为分钟，0表示只告警一次
AI2告警周期		循环告警的时间间隔，单位为分钟，0表示只告警一次
AI1低于下限告警短信内容		最多30个汉字或60个字符
AI1恢复短信内容		最多30个汉字或60个字符
AI1超出上限告警短信内容		最多30个汉字或60个字符
AI2低于下限告警短信内容		最多30个汉字或60个字符
AI2恢复短信内容		最多30个汉字或60个字符
AI2超出上限告警短信内容		最多30个汉字或60个字符

告警内容案例：

配置如下：

安装地址：“北厂区：”

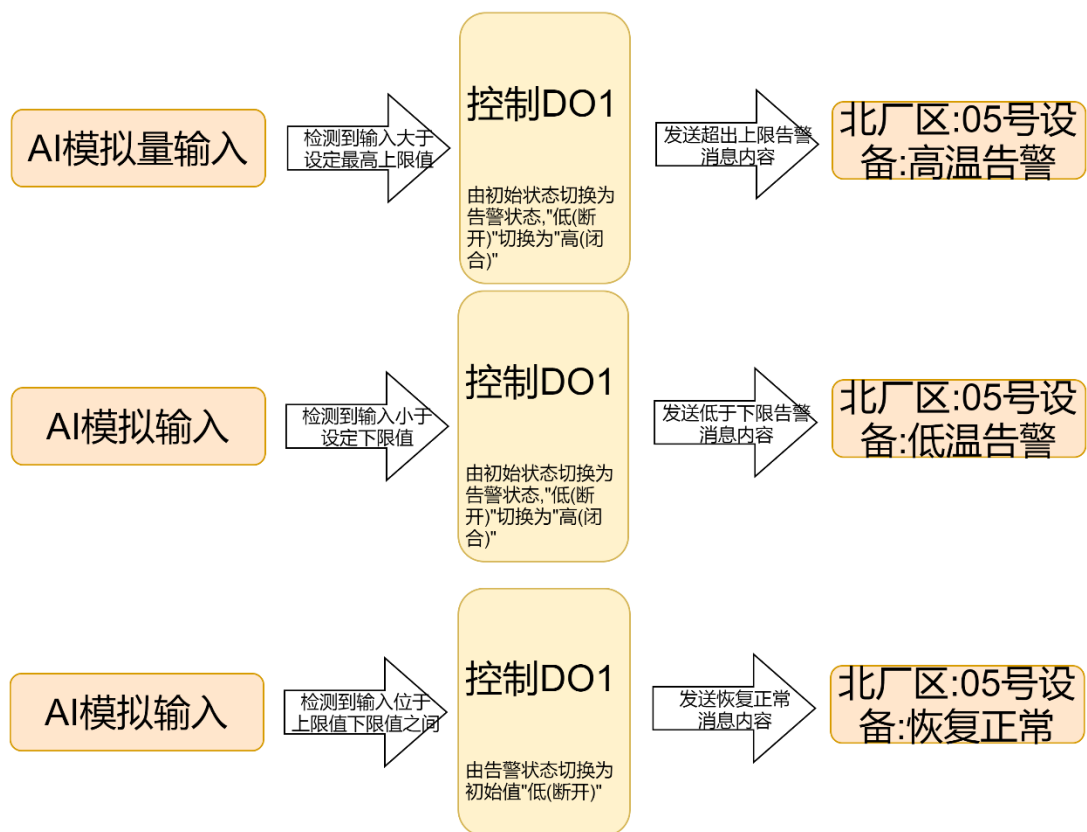
设备身份 ID：“05 号设备：”

AI1 低于下限值短信内容：“低温警告”

<短信告警内容参数>		当告警通道设置为短信时，此项参数才有效
安装地址	北厂区：	最长16个汉字,32个字符，短信报警用
设备身份ID	05号设备：	8位编码，短信报警时有效
AI1告警周期	1	循环告警的时间间隔，单位为分钟，0表示只告警一次
AI2告警周期	1	循环告警的时间间隔，单位为分钟，0表示只告警一次
AI1低于下限告警短信内容	低温告警	最多30个汉字或60个字符
AI1恢复短信内容	恢复正常水平	最多30个汉字或60个字符
AI1超出上限告警短信内容	高温告警	最多30个汉字或60个字符

其他告警短信内容格式与低于下限值格式一致（消息内容为空则不发送内容）

短信告警流程如下



### 5.6.3 RTU 数据可视化测试

参数配置软件包含有 RTU 数据可视化功能, 可实现简单的模块功能测试。  
操作步骤如下图:



- 选择“RTU 可视化”界面
- 配置 USB 转 485 通讯参数并打开串口
- 设置 Modbus 子站地址为默认“1”
- 点选“自动读取”复选框

注意：此时界面左侧仪表盘会根据模拟量信号源的输出显示对应电流读数，右侧会有上下行通讯报文。若无上行数据请检查 USB-485 转换器是否正常工作，或尝试模块 485 AB 之间接 120 欧电阻。

详细的测试演示可参考[“用户测试文档”](#)

六、协议详解

地址域	功能码	数据	差错检验
-----	-----	----	------

Modbus 使用“big-Endian”（大端模式）表示地址和数据项，这就意味着当发射多个字节时，首先发送最高字节。

例如：寄存器地址为 0x0014，首先发送的是 0x00，然后才是 0x14。

一个正常的 Modbus 响应：响应功能码=请求功能码。

一个 Modbus 的异常响应：响应功能码=请求功能码+0x80，提供一个异常码来指示差错原因。

6.1 功能码描述

6.1.1 01 读线圈

可以使用此功能码读取继电器 DOx 的状态。

请求 PDU 详细说明了起始地址，即指定第一个线圈的地址和线圈数量，从零开始寻址线圈，因此寻址线圈 1-N 为 0-(N-1)。

响应 PDU 中 N 个字节的线圈状态的每一个 bit 位代表一个线圈的状态，状态 1=ON，0=OFF。第一个字节的最低位 LSB 代表第 0 号线圈的状态（即起始地址指定的线圈号为 0 号线圈），其他线圈依次类推，一直到这个字节的最高位 MSB 为止，并且后续字节中都是由低到高代表连续的各线圈状态。

如果线圈数量不是 8 的倍数，将用零填充剩余最后数据字节中的剩余比特，字节数量域说明了数据的完整字节数。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x01
起始地址	2 个字节	0x0014 至 0x0015
线圈数量	2 个字节	n(1 至 n-1)
CRC 校验	2 个字节	

注：线圈状态的字节数 N=线圈数量 n/8，如果余数不等于 0，则 N=n/8+1

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x81 （请求功能码+0x80）
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个读离散量 DO1 的实例

请求	响应
----	----

地址	01	地址	01
功能码	01	功能码	01
起始地址高 H	00	字节数	01
起始地址低 L	14	D01-D04 状态	01
线圈数量高 H	00	CRC 校验高 H	90
线圈数量低 L	01	CRC 校验低 L	48
CRC 校验高 H	BD		
CRC 校验低 L	CE		

发送：010100140001BDCE      RTU 响应：010101019048

D01 的状态字节为 0D，二进制 00000001，D01 是这个字节的 LSB(第 0 位)为 1 表示闭合，其他 D0x 是第(x-1)位为 0 表示断开，用 0 填充未使用位。

### 6.1.2 03 读保持寄存器/04 读输入寄存器

使用该功能码可以读取所有寄存器包括 AIx、DOx、DIx 的状态。

请求 PDU 详细说明了起始寄存器地址和寄存器数量，从 0 开始寻址寄存器，因此寻址寄存器 1-N 为 0-(N-1)。

响应报文中的寄存器数据每个寄存器有 2 个字节，对于每一个寄存器，第一个字节代表寄存器值的高位，第二个字节代表寄存器值的低位。字节数为寄存器数量乘以 2。对于 AI，一个通道占用 2 个寄存器，4 个字节的值使用浮点数表示，对于 DOx，2 个字节的值 0000 代表继电器断开，0001 代表继电器闭合，对于 DIx，2 个字节的值 0000 代表开关量无输入，0001 代表有输入。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x03 或 04
起始地址	2 个字节	0x0000 至 0x0017
寄存器数量	2 个字节	n(1 至 N)
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x03 或 0x04
字节数	1 个字节	N=2*n
寄存器值	N 个字节	N=2*n, n 为寄存器数量
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x83 或 0x84 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04

CRC 校验	2 个字节	
--------	-------	--

这是一个读模拟量输入 AI1 的实例

请求		响应	
地址	01	地址	01
功能码	03	功能码	03
起始地址高 H	00	字节数	04
起始地址低 L	00	AI1 值	4 个字节
寄存器数量高 H	00	CRC 校验高 H	
寄存器数量低 L	02	CRC 校验低 L	
CRC 校验高 H	C4		
CRC 校验低 L	0B		

发送: 010300000002C40B RTU 响应:0103044019999AD40F

### 6.1.3 05 写单个线圈

可以使用该功能码写单个继电器 DOx 为断开或闭合

请求数据域中的常量说明请求的 ON/OFF 状态，十六进制值 0xFF00 请求输出为 ON(闭合)，十六进制值 0x0000 请求输出为 OFF(断开)，其他所有值都是非法的，对输出不起作用，RTU 返回错误响应。

请求域中的输出地址规定了要写入线圈的地址。

正常响应是请求的应答，在写入线圈状态后返回这个正常响应。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x05
输出地址	2 个字节	0x0014 至 0x0015
输出值	2 个字节	0x0000 或 0xFF00
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x05
输出地址	2 个字节	0x0014 至 0x0015
输出值	2 个字节	0x0000 或 0xFF00
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x85 (请求功能码+0x80)

异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个请求写线圈 D01 为 ON(闭合)的实例

请求		响应	
地址	01	地址	01
功能码	05	功能码	05
输出地址高 H	00	输出地址高 H	00
输出地址低 L	14	输出地址低 L	14
输出值高 H	FF	输出值高 H	FF
输出值低 L	00	输出值低 L	00
CRC 校验高 H	CC	CRC 校验高 H	CC
CRC 校验低 L	3E	CRC 校验低 L	3E

发送：01050014FF00CC3E

RTU 响应：01050014FF00CC3E

#### 6.1.4 06 写单个寄存器

可以使用该功能码写单个继电器 DOx 为断开或闭合。

请求数据域中的寄存器值说明请求的 ON/OFF 状态，十六进制值 0001 请求输出为 ON(闭合)，十六进制值 0x0000 请求输出为 OFF(断开)。

请求域中的寄存器地址规定了要写入线圈的地址。

正常响应是请求的应答，在写入线圈状态后返回这个正常响应。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x06
寄存器地址	2 个字节	0x0014 至 0x0015
寄存器值	2 个字节	0x0000 至 0xFFFF
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x06
寄存器地址	2 个字节	0x0014 至 0x0015
寄存器值	2 个字节	0x0000 至 0xFFFF
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x86 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04

CRC 校验	2 个字节	
--------	-------	--

这是一个请求写线圈 D01 为 ON(闭合)的实例

请求		响应	
地址	01	地址	01
功能码	06	功能码	06
寄存器地址高 H	00	寄存器地址高 H	00
寄存器地址低 L	14	寄存器地址低 L	14
寄存器值高 H	00	寄存器值高 H	00
寄存器值低 L	01	寄存器值低 L	01
CRC 校验高 H	08	CRC 校验高 H	08
CRC 校验低 L	0E	CRC 校验低 L	0E

发送：010600140001080E

RTU 响应：010600140001080E

### 6.1.5 0F 写多个线圈

可以使用此功能码写多个继电器 DOx 为断开或闭合。

请求 PDU 详细说明了起始地址，即指定第一个线圈的地址和线圈数量，从零开始寻址线圈，因此寻址线圈 1-N 为 0-(N-1)。

请求数据域中的内容说明了被请求的 ON/OFF 状态，域比特位中的逻辑“1”请求相应输出为 ON，域比特位中的逻辑“0”请求相应输出为 OFF。从数据域中第一个字节的 bit0 开始到 bit7，然后到第二个字节的 bit0，依次表示第一个线圈到第 n 个线圈的 ON/OFF 值。

正常响应返回功能码、起始地址和线圈数量。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x0F
起始地址	2 个字节	0x0014 至 0x0015
线圈数量	2 个字节	n(1 至 N)
字节数	1 个字节	$N=n/8$ , 或 $N=n/8+1$
输出值	N 个字节	
CRC 校验	2 个字节	

注：线圈输出字节数  $N=$ 线圈数量  $n/8$ ，如果余数不等于 0，则  $N=n/8+1$

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x0F
起始地址	2 个字节	0x0014 至 0x0015
线圈数量	2 个字节	n(1 至 2)
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x8F (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个请求从线圈 D01 开始写入 1 个线圈的实例

请求		响应	
地址	01	地址	01
功能码	0F	功能码	0F
起始地址高 H	00	起始地址高 H	00
起始地址低 L	14	起始地址低 L	14
线圈数量高 H	00	线圈数量高 H	00
线圈数量低 L	01	线圈数量低 L	01
字节数	01	CRC 校验高 H	D4
输出值	01	CRC 校验低 L	0F
CRC 校验高 H	DF		
CRC 校验低 L	54		

发送：010F0014000201012F51      RTU 响应：010F00140001D40F

D01 的输出值为 01，二进制 00000001，D01 是这个字节的 LSB(第 0 位)为 0 表示断开，D0x 是第 (x-1) 位为 1 表示闭合，用 0 填充剩余未使用位。

### 6.1.6 10 写多个寄存器

使用该功能码可以写连续寄存器 D0x 的状态。

请求 PDU 详细说明了起始寄存器地址、寄存器数量、字节数和寄存器值，从零开始寻址寄存器，因此寻址寄存器 1-N 为 0-(N-1)。

寄存器数据中每个寄存器有 2 个字节，对于每一个寄存器，第一个字节代表寄存器值的高位，第二个字节代表寄存器值的低位。字节数为寄存器数量乘以 2，2 个字节的值 0000 代表继电器断开，0001 代表继电器闭合。

正常响应返回功能码、起始地址和被写入寄存器的数量。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x10
起始地址	2 个字节	0x0014 至 0x0015
寄存器数量	2 个字节	n (1 至 N)
字节数	1 个字节	N=2*n
寄存器值	N 个字节	N=2*n,    n 为寄存器数量
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x10
起始地址	2 个字节	0x0014 至 0x0015
寄存器数量	2 个字节	n(1 至 2)
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x90 （请求功能码+0x80）
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个控制继电器 DOx 的实例

请求		响应	
地址	01	地址	01
功能码	10	功能码	10
起始地址高 H	00	起始地址高 H	00
起始地址低 L	14	起始地址低 L	14
寄存器数量高 H	00	寄存器数量高 H	00
寄存器数量低 L	01	寄存器数量低 L	01
字节数	02	CRC 校验高 H	41
DO1 寄存器值高 H	00	CRC 校验低 L	CD
DO1 寄存器值高 L	01		
CRC 校验高 H	64		
CRC 校验低 L	84		

发送：0110001400010200016484      RTU 响应：01100014000141CD

DO1 寄存器值为 0001 表示闭合

## 6.2 错误码描述

错误码含义：当 DTU 收到错误的 Modbus 指令时，会返回功能码为请求功能码+0x80，紧随着一个字节的错误码代表出错原因。

错误码 01：表示不支持的功能码，众山 DTU 支持上述 8 种功能码，除此之外的功能码都会返回错误码为 01 的错误。

错误码 02：表示起始地址不存在或者起始地址加上寄存器数量后的地址不存在。总的来说表示访问的寄存器不存在。

错误码 03：表示寄存器数量不符合规范或者寄存器值非法。

错误码 04：表示读写寄存器错误。

## 6.3 CRC 校验算法

CRC 即[循环冗余校验码](#) (Cyclic Redundancy Check)：是数据通信领域中最常用的一种查错校验码，其特征是信息字段和校验字段的长度可以任意选定。循环冗余检查 (CRC) 是一种数据传输检错功能，对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

ModbusRTU 的 CRC16 计算初值：0xFFFF

ModbusRTU 的 CRC16 计算多项式 0xA001 (二进制:1010 0000 0000 0001)

附 CRC 校验算法代码：

```
uint16_t mb_crc( uint8_t* snd, uint16_t num )
{
    uint8_t CRC_Lb, CRC_Hb;
    uint8_t CRC_L, CRC_H;
    uint16_t crc;

    CRC_H = 0xFF;
    CRC_L = 0xFF;

    for ( uint16_t i = 0; i < num; i++ ) {
        CRC_L = CRC_L ^ snd[ i ];
        for ( uint16_t j = 0; j < 8; j++ ) {
            CRC_Lb = CRC_L;
            if ( ( CRC_L & 1 ) == 1 ) {
                CRC_L = ( CRC_L - 1 ) / 2;
                CRC_Lb = CRC_L;
                CRC_Hb = CRC_H;
                if ( ( CRC_H & 1 ) == 1 ) {
                    CRC_L = CRC_L + 128;
                    CRC_Lb = CRC_L;
                    CRC_H = ( CRC_H - 1 ) / 2;
                    CRC_Hb = CRC_H;
                } else {
                    CRC_H = CRC_H / 2;
                    CRC_Hb = CRC_H;
                }
            }
            CRC_L = CRC_L ^ 1;
            CRC_Lb = CRC_L;
            CRC_H = CRC_H ^ 0xA0;
            CRC_Hb = CRC_H;
        } else {
            CRC_L = CRC_L / 2;
        }
    }
```

```
CRC_Lb = CRC_L;
CRC_Hb = CRC_H;
if ( ( CRC_H & 1 ) == 1 ) {
    CRC_L = CRC_L + 128;
    CRC_Lb = CRC_L;
    CRC_H = ( CRC_H - 1 ) / 2;
    CRC_Hb = CRC_H;
} else {
    CRC_H = CRC_H / 2;
    CRC_Hb = CRC_H;
}
}

crc = CRC_L;
crc <<= 8;
crc |= CRC_H;
return crc;
}
```

## 七、更改记录

v1.6

- \* 新增 AI 整数段寄存器读取说明以及示例

v1.5

- \* 更新有源、无源接线方式

v1.4

- \* 更新外壳尺寸图
- \* 更新电池检测接线方式

v1.3

- \* 接线图新增三线制

v1.2

- \* 修改接线图，调整文字字体

v1.1

- \* 修改 3.1 接线图，3.2 跳线图，3.3LED 指示灯配图

v1.0

- \* 第一版编写